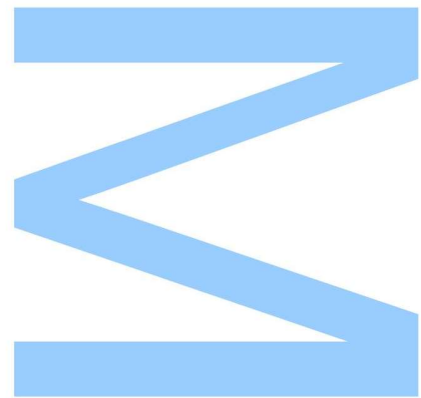




D&C SOFTWARE, LDA

Aplicação em dispositivos móveis para empresa de Transporte de Mercadorias



Hugo Miguel Figueiredo de Oliveira

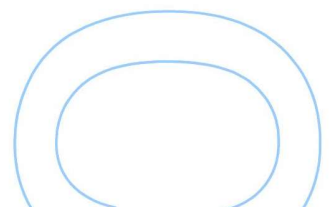
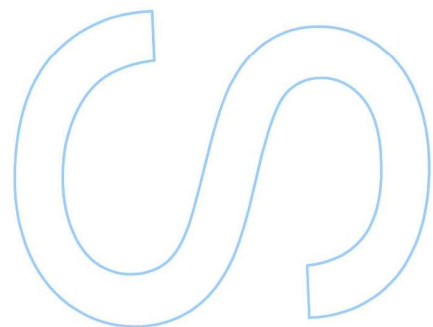
Mestrado Integrado em Engenharia de Redes e Sistemas Informáticos
Departamento de Ciência de Computadores
2016

Orientador

Eng.º José Costa, D&C Software Lda.

Coorientador

Prof. Rita Ribeiro, DCC-FCUP

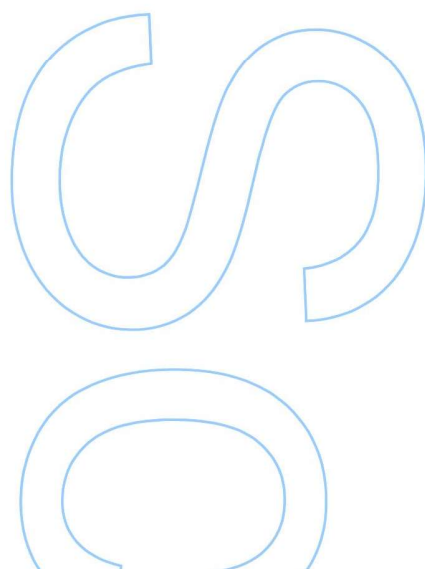
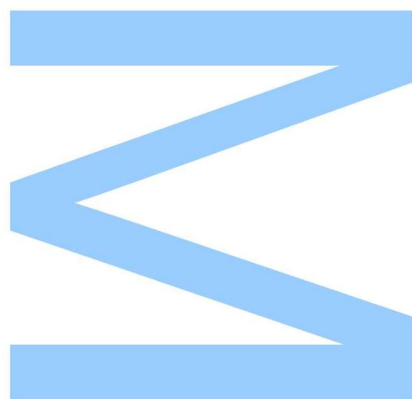




Todas as correções determinadas
pelo júri, e só essas, foram efetuadas.

O Presidente do Júri,

Porto, ____/____/____



”Persistence is the shortest path to success!”
Charles Chaplin

Agradecimentos

Queria começar por agradecer há minha família e amigos que direta e indiretamente me ajudaram durante todo o meu percurso académico, sendo sempre um modelo de inspiração, dando ajuda, apoio incondicional, motivação e nunca deixando que desistir fosse uma opção.

Gostaria de agradecer à empresa D&C Software, em especial ao Eng.º José Costa e Eng.º José Nuno Costa pela oportunidade e confiança demonstrada, bem como por toda a disponibilidade e preocupação em resolver todos os meus assuntos relacionados com o estágio.

Agradeço também à minha co-orientadora, Professora Rita Ribeiro, pela ajuda dada a escrever este relatório e por toda a disponibilidade e paciência demonstrada durante todo este projeto.

Por último, mas não menos importante, gostaria de agradecer a toda Faculdade Ciências em especial a todo o corpo docente do Departamento de Ciências dos Computadores, por me terem formado e orientado da melhor forma durante todos os anos de curso, preparando para o mundo profissional.

O meu mais sincero agradecimento a todos.

Resumo

Cada vez mais, as novas tecnologias têm tido um papel preponderante na evolução da indústria, tendo, de dia para dia, vindo a surgir projetos com a finalidade de facilitar e melhorar a forma como as empresas trabalham. O transporte de mercadoria, sendo um setor de grande importância para o comércio global, tem que seguir este desenvolvimento e tirar partido do que as novas tecnologias têm para oferecer.

Foi seguindo esta perspetiva de evolução que a empresa D&C Software surgiu com a ideia de utilizar os dispositivos móveis que têm vindo a tornar-se uma ferramenta indispensável no nosso dia-a-dia, e desenvolver a Transportes App. Uma aplicação para *smartphones* que tem como objetivo, tirar partido do grande poder dos dispositivos móveis, conciliando os trabalhos desempenhados pelos motoristas de transportes de mercadorias. Isto é uma mais-valia para as empresas, visto que passam a possuir um novo canal de comunicação com os seus motoristas que possibilita o acompanhamento em tempo real dos seus serviços, para uma melhor gestão dos mesmos.

Para começar o projeto foi feito um levantamento de algumas funcionalidades que podiam ser interessantes serem implementadas, bem como o desenvolvimento de uma maquete para apresentar a possíveis utilizadores. O *feedback* recebido foi usado como guia para a elaboração do projeto.

No presente relatório, são apresentadas todas as funcionalidades implementadas na aplicação, bem como as ferramentas e tecnologias usadas no seu desenvolvimento. Identificando as dificuldades encontradas e sugerindo trabalho futuro que pode ser feito com o intuito de melhorar esta plataforma.

Palavras-chave. D&C Software, Transporte de Mercadorias, *Smartphones*, Android, Node.js, PostgreSQL

Conteúdo

Resumo	iii
Lista de Tabelas	vii
Lista de Figuras	viii
1 Introdução	1
1.1 Motivação	1
1.2 Objetivos	2
1.3 Organização do Relatório	2
2 Enquadramento	3
2.1 Empresas de Transporte de mercadorias	3
2.1.1 Serviços	4
2.1.2 Preocupações	5
2.2 Aplicações Existentes	5
2.2.1 LSNet	5
2.2.2 Wtransnet Cargo	7
2.2.3 Comparação	9
3 Transportes App	10

3.1	Requisitos	10
3.2	Funcionalidades	11
3.3	Arquitetura	12
3.3.1	Ligação Cliente-Servidor	13
3.3.2	Ligação Base de Dados	14
3.4	Tecnologias	14
3.4.1	Android	14
3.4.1.1	Android Studio	16
3.4.1.2	Desenvolvimento de aplicações Android	16
3.4.2	Node.js	18
3.4.2.1	Módulos	19
3.4.2.2	WebStorm	19
3.4.3	PostgreSQL	20
3.4.3.1	pgAdmin	20
4	Implementação do Sistema	21
4.1	Base de Dados	21
4.1.1	Servidor	21
4.1.2	Cliente	23
4.2	API REST	24
4.2.1	Rotas	24
4.2.2	Tokens	26
4.2.3	Ligação à Base de Dados	27
4.2.4	Guia de Transporte	27
4.2.5	Fotografias	28
4.3	Aplicação Android	28

4.3.1	Login	28
4.3.2	Descrição do Serviço	29
4.3.3	Serviço	30
4.3.4	Anexos	32
4.3.5	Despesas	33
4.3.6	Guia de Transporte	34
4.3.7	Barra de menu	34
4.3.8	Histórico	35
4.3.9	Visualizar guias de Transporte	36
5	Conclusão	38
5.1	Contribuições e satisfação dos objetivos	38
5.2	Dificuldades encontradas	39
5.3	Trabalho futuro	39
A	Acrónimos	40
	Referências	40

Lista de Tabelas

2.1	Tabela comparativa de versões das aplicações LSNet e Wtransnet Cargo.	9
3.1	Versões <i>Android</i>	11
4.1	Tabela de rotas	26

Lista de Figuras

2.1	Aplicação LSNet. [35]	6
2.1	Aplicação LSNet.(cont.) [35]	7
2.2	Aplicação Wtransnet Cargo. [41]	8
3.1	Arquitetura de 3 camadas.	13
3.2	Logo <i>Android</i> .	15
3.3	Logo <i>Node.js</i> . [29]	18
3.4	Logo <i>PostgresSql.js</i> [40].	20
4.1	Base de Dados Servidor.	22
4.2	Base de Dados Cliente.	23
4.3	Transportes App - Login	29
4.4	Transportes App - Descrição do serviço	30
4.5	Transportes App - Serviço a decorrer	31
4.6	Transportes App - Anexos	32
4.7	Transportes App - Despesas	33
4.8	Transportes App - Criar Guia de transporte	34
4.9	Transportes App - Barra de menu	35
4.10	Transportes App - Histórico de serviços	36
4.11	Transportes App - Guias de Transporte	37

Capítulo 1

Introdução

Com o decorrer dos tempos, o grande avanço tecnológico nos sistemas de transportes e de comunicação tem sido evidente. Tendo estimulado o aparecimento de inúmeras inovações nos diversos setores industriais, com o propósito de facilitar e melhorar a qualidade dos trabalhos e produtos produzidos pelas empresas.

1.1 Motivação

A oportunidade de integrar a equipa de desenvolvimento da empresa D&C Software revelou-se uma mais-valia e um desafio gratificante. Esta experiência permitiu-me aplicar competências adquiridas durante o meu percurso académico e, possibilitou o meu contributo no desenvolvimento de uma aplicação direcionada ao setor de transporte de mercadorias, setor esse tão importante para a economia global.

Tirando partido da grande popularidade dos *smartphones* e dispositivos moveis que incorporam as mais diversas ferramentas que podem ser utilizadas para simplificar e agilizar as mais diversas tarefas, surgiu a criação da aplicação Transportes app.

Uma aplicação que tem como grande motivação, procurar ser uma mais valia para os motoristas, ajudando-os a desempenhar o seu trabalho de uma forma mais organizada e simples.

1.2 Objetivos

Os objetivos deste projeto passam pela criação de uma aplicação para *smartphones* que permita aos motoristas de uma empresa de transportes de mercadorias, receber serviços a desempenhar e tirando partido de varias funcionalidades da aplicação, poder registar em tempo real informações como hora, local, quilómetros, peso aproximado da carga, despesas contraídas no serviço e até criação de uma guia de transporte.

Com a preocupação que a interface seja simples e intuitiva, esta aplicação tem o propósito de facilitar não só o trabalho do motorista mas também a gestão de serviços por parte do departamento de logística, que têm agora o poder de atribuir e acompanhar serviços em tempo real.

1.3 Organização do Relatório

A organização deste relatório segue a seguinte estrutura:

O Capítulo 2, é destinado a introduzir os leitores ao setor dos transportes de mercadorias, onde são descritos os vários tipos de transportes de mercadorias bem como as suas vantagens e desvantagens, os vários tipos de serviços existentes e até certas preocupações que têm vindo a afligir este setor. Conclui-se o capítulo com uma análise a duas aplicações para *smartphones* relacionadas com este setor de atividade.

No Capítulo 3, é feita uma introdução à aplicação Transportes App, onde são descritos alguns requisitos da aplicação e onde é feito um levantamento e análise das tecnologias utilizadas no decorrer do projeto.

No Capítulo 4, é feita uma análise mais pormenorizada da aplicação, descrevendo toda a sua implementação, concluindo com uma visão sobre alguns dos ecrãs que fazem parte de esta aplicação.

Por fim, no Capítulo 5 são apresentadas as conclusões do trabalho realizado, bem como algumas sugestões para algum trabalho futuro a ser feito com o intuito de melhorar a qualidade do projeto.

Capítulo 2

Enquadramento

2.1 Empresas de Transporte de mercadorias

O transporte de mercadorias é um setor que se encontra em constante evolução e crescimento, tendo uma elevada importância no movimento da economia mundial [26]. Em 2015, segundo a Associação Nacional de Transportadores Públicos ([ATRAM](#)) contávamos com cerca de 8 mil empresas de transporte de mercadorias em Portugal [9].

Esta atividade consiste na deslocação física de mercadorias ou produtos de diversos tipos, recorrendo a diferentes meios de transporte: marítimo, aéreo, ferroviário e rodoviário.

O transporte marítimo tem como vantagem conseguir transportar cargas de grandes dimensões e até elevadas quantidades. Porém, com o inconveniente de ser de baixa velocidade e pouco flexível [33].

O transporte aéreo, sendo muito rápido e preferível para mercadorias de pouca dimensão, tem a desvantagem de ser muito custoso e limitado na quantidade de carga.

O transporte ferroviário tem como pontos positivos o menor custo de transporte para grandes distâncias, não sendo afetado por condições climatéricas permitindo uma quantidade considerável de carga. Os seus pontos fracos residem na sua pouca flexibilidade em trajetos.

Por último, temos os transportes rodoviários, aqueles que serão abordados no presente relatório, têm a particularidade de possuírem uma rede muito vasta de trajetos tornando-os muito flexíveis a nível de itinerários. São bastante práticos e económicos

em viagens de curta ou média distancia.

Podemos marcar como inconvenientes serem: diretamente influenciados pela quantidade de tráfego local existente, influenciando, dessa forma, o tempo despendido no serviço; exigirem uma contínua construção e manutenção das vias por onde circulam; e ainda serem limitados na quantidade de mercadorias a transportar, isto é, de conseguirem transportar consideravelmente menos mercadorias comparativamente a outros meios de transporte, como ferroviário e marítimo [21].

Para além de ser um exercício fulcral para o comércio mundial, este sector oferece um número considerável de postos de trabalho tendo em conta ao elevado número de empresas existentes. Podemos dividir os funcionários destas empresas em dois grupos. Os funcionários que passam a maior parte do seu tempo no edifício central da empresa a tratarem de toda a logística, contabilidade e administração relacionada com a empresa e os serviços prestados. No outro lado, temos os motoristas que estão diretamente envolvidos no ato de transporte das mercadorias, conduzindo os veículos pelos pontos de carga e descarga dos produtos [17].

Dependendo do tamanho destas empresas e do número de serviços a prestar, a quantidade de funções desempenhadas por cada funcionário poderá aumentar ou diminuir, existindo mesmo muitas vezes a necessidade de subcontratar motoristas para conseguir atender ao elevado número de serviços requisitados. Existem casos em que se procede ao aluguer de veículos que se encontram sem serviço, de forma rentabilizar o elevado número de veículos para a pouca quantidade de trabalho.

2.1.1 Serviços

Os serviços desempenhados por estas entidades podem ser classificados geograficamente como transporte local, nacional ou até mesmo internacional, tendo em consideração o ponto de origem e de destino do respetivo serviço.

O transporte de mercadorias de natureza local está relacionado com o movimento das mercadorias a curta distância, por outras palavras, dentro da mesma cidade ou região. A nível nacional, temos o transporte dos produtos dentro do mesmo país. Por último, temos o transporte internacional, quando este ocorre entre países (ou continentes).

Dentro dos serviços efetuados entre países, estes podem ser divididos em dois tipos: importação e exportação. Importação refere-se à entrada no país de uma carga proveniente de um país estrangeiro. Exportação refere-se à saída de mercadoria para

fora do país [18].

2.1.2 Preocupações

Sendo um setor muito influenciado pela economia, as empresas de transporte de mercadorias sentem o efeito de recessão e da quebra da atividade económica. Em 2010, a ATRAM noticiou que, devido à crise, cerca de 1500 empresas terão fechado portas. Os motivos vão desde o aumento do preço dos combustíveis até ao aumento do preço e número de portagens, aumento da carga fiscal, entre outros [9].

Deparadas com esta situação, estas empresas vêm-se obrigadas a realizar uma gestão logística eficiente, planeando com atenção serviços, controlando mais atentamente os seus motoristas de forma a maximizar os serviços prestados e minimizar os custos a eles associados. Existe uma preocupação de criar soluções atrativas e adaptadas aos clientes, tornando a empresa mais competitiva no mercado [17].

2.2 Aplicações Existentes

As aplicações que se seguem são dois exemplos de aplicações para *smarthphones* e *tablets*, que estão relacionadas com o setor de transporte de mercadorias. Foi feito um levantamento de algumas aplicações utilizadas no mercado português, e após uma análise, estas foram escolhidas por serem direcionadas a dois tipos de utilizadores diferentes: clientes das transportadoras que requisitam serviços e as próprias empresas de transporte de mercadorias que procuram serviços. O objetivo é dar uma visão mais diversificada das aplicações.

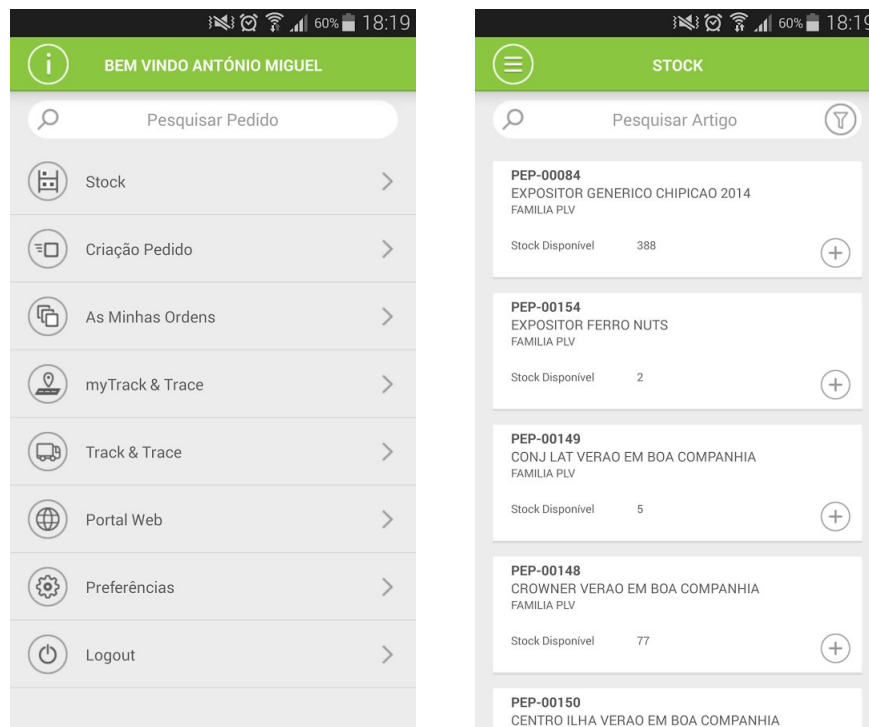
2.2.1 LSNet

LSNet [1] foi lançada em 2015, por um dos maiores grupos de transportadoras a nível nacional, o grupo Luís Simões, com o intuito de melhorar o seu serviço e de se tornar mais competitiva, recorrendo às novas tecnologias [19]. Está disponível para os sistemas *Android* e *IOS*, podendo o seu *download* ser feito tanto na loja, *Google Play Store* no caso de aparelhos *Android*, como na *App Store* para dispositivos *Apple*.

Esta aplicação é direcionada aos clientes da transportadora, que poderão através de qualquer dispositivo móvel criar pedidos e aceitar ordens de entrega, sendo automati-

camente integradas nos sistemas da empresa [37]. Estes pedidos poderão também ser acompanhados ao longo de todo o seu processo através do sistema *Track & Trace*. A aplicação possibilita ainda a consulta de stocks, de forma detalhada e dinâmica além de toda a atividade ao longo de um ano [34].

A Figura 2.1a mostra o menu principal da aplicação LSNet, onde podemos verificar que se trata de uma aplicação bastante simples e intuitiva. Neste ecrã é nos apresentado a opção para consultar os stocks (cf. Figura 2.1b), criar pedidos no ponto de venda (cf. Figura 2.1c), o *Track & Trace* (cf. Figura 2.1d) entre outros.



(a) Ecrã de menu.

(b) Ecrã de stocks.

Figura 2.1: Aplicação LSNet. [35]

A LSNet torna-se bastante vantajosa visto que proporciona aos clientes da empresa Luís Simões maior autonomia na sua gestão de mercadorias, reduzindo a sua carga administrativa.

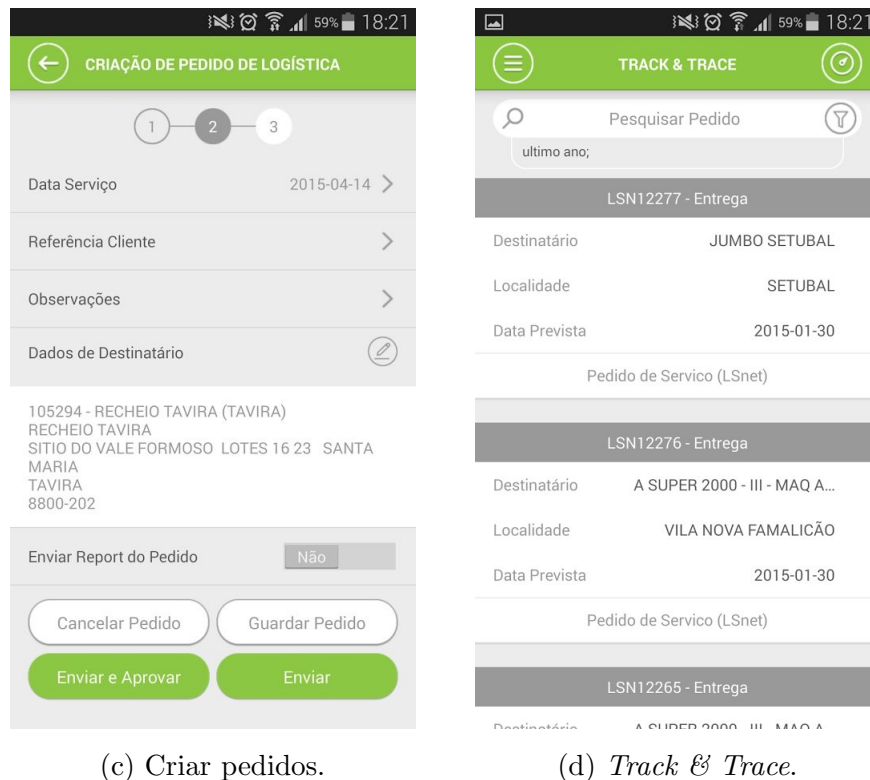


Figura 2.1: Aplicação LSNet.(cont.) [35]

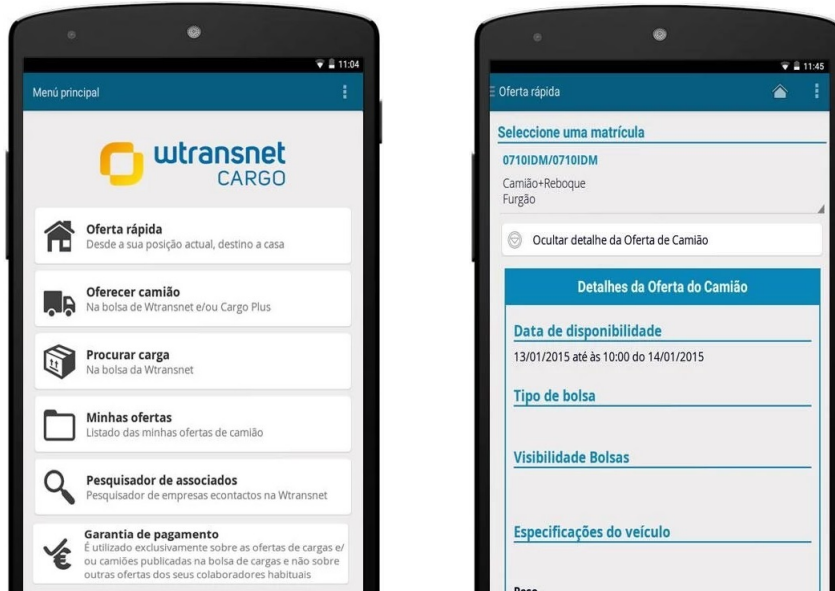
2.2.2 Wtransnet Cargo

A Wtransnet Cargo [2] é uma aplicação de *smartphone*, disponibilizada em 2015 pela Wtransnet somente a dispositivos com o sistema *Android* no *Google Play Store*. Wtransnet nasceu em 1996 como uma bolsa de cargas de camião. Com o passar do tempo foi-se convertendo numa plataforma de *networking* do sector do transporte e logística. Com sede em Espanha, conta com uma equipa de 125 elementos de 8 nacionalidades diferentes.

Direcionada a todas as empresas de transportes e transportadores autónomos [38], esta ferramenta permite publicitar, facilmente, camiões disponíveis, localizar através do sistema de geolocalização cargas em toda a Europa, que se encontrem no trajeto definido pelo utilizador e até a possibilidade de efetuar pagamentos e emitir faturas pela aplicação [41].

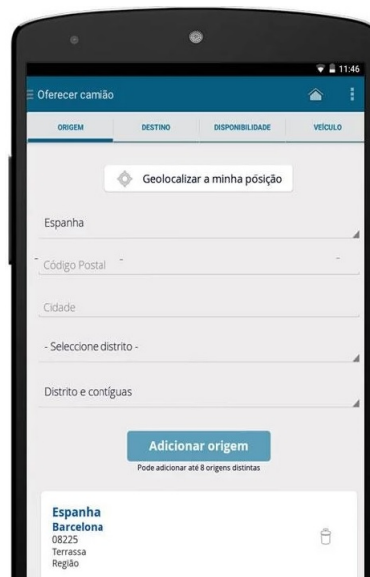
Após ser feito o *login* na aplicação, surge o ecrã principal como apresentado na Figura 2.2a, onde temos algumas das opções já mencionadas anteriormente. A oferta de um camião poderá ser feita preenchendo os campos apresentados na Figura 2.2b ou então

pelo formulário da fig 2.2c, que nos oferece possibilidade de definir uma origem e destino de um trajeto, com o intuito de verificar se existe alguma carga disponível no trajeto entre os pontos introduzidos.



(a) Ecrã de menu.

(b) Disponibilizar camião.



(c) Planear trajeto

Figura 2.2: Aplicação Wtransnet Cargo. [41]

Esta ferramenta é como a empresa afirma, “(...) desenhada a pensar nas necessidades diárias dos trabalhadores autónomos e empresas de transporte(...)” [38]. Oferece um local organizado e centralizado de informação relevante para todas estas entidades.

Um requisito imposto à utilização desta plataforma, tanto às empresas como aos trabalhadores singulares, é a necessidade de se associarem à rede Wtransnet. Este registo poderá ser efetuado através do *website* da instituição.

2.2.3 Comparação

Apesar de ambos os exemplos referidos serem destinados ao setor de transporte de mercadorias, estas aplicações foram criadas com diferentes objetivos.

No caso da LSNet, a empresa Luís Simões tem como principal foco agilizar a comunicação dos seus clientes com os fornecedores e, dessa forma, potenciar o seu serviço de transporte e logística [37]. Por outro lado, a aplicação disponibilizada pela Wtransnet tem como finalidade a criação de uma rede onde todas as empresas e transportadores autónomos possam publicitar os seus serviços.

Ambas as aplicações estão disponíveis para o sistema *Android*. A LSNet também é compatível com sistemas *IOS*. As duas aplicações são gratuitas e impõem a obrigatoriedade aos utilizadores de possuírem uma conta de utilizador que pode ser criada no *website* da respetiva empresa.

Aplicações	<i>Android</i>	<i>IOS</i>	Versão
LSNet	$\geq 4.0.3$	≥ 7.0	1.1.3
Wtransnet Cargo	≥ 4.1	N/A	1.1.1

Tabela 2.1: Tabela comparativa de versões das aplicações LSNet e Wtransnet Cargo.

Capítulo 3

Transportes App

A Transportes App, aplicação desenvolvida no âmbito deste projeto, à semelhança das aplicações referidas no capítulo anterior, destina-se ao setor de transportes de mercadorias tendo sido desenvolvida para *smartphones*. Por outro lado, enquanto as aplicações mencionadas se destinam a clientes, a Transportes App é para uso interno. Destina-se exclusivamente aos funcionários de uma empresa de transporte, mais especificamente, para os seus camionistas.

Os camionistas poderão através da aplicação receber serviços, com locais de carga e descarga associados. Ao longo do serviço poderão registar informações relativas a cada uma das fases. Desta forma, funcionários no escritório que tratam de toda logística associada, possam acompanhar o desenrolar do serviço.

3.1 Requisitos

Os principais requisitos impostos no desenvolvimento da aplicação foram: a compatibilidade com o sistema *Android* e garantir a compatibilidade com um número considerável de *smartphones* vendidos nos dias de hoje. Tendo em conta estes fatores, a versão de *Android* considerada mínima para que um dispositivo possa correr a aplicação, é a 5.x. Isto porque se trata de uma versão relativamente recente e como podemos verificar pela Tabela 3.1 já consegue cobrir cerca de 45% dos dispositivos, valor que tende a aumentar com o passar do tempo.

Para além da versão do *Android*, seria importante que o *smartphone* dispusesse das seguintes características para que se possa tirar partido de todas as funcionalidades

Versão	Nome	API	Distribuição	Lançamento
2.2	Froyo	8	0,1%	20 de Maio, 2010
2.3.3 – 2.3.7	Gingerbread	10	2,0%	9 de Fevereiro, 2011
4.0.3 – 4.0.4	Ice Cream Sandwich	15	1,9%	16 de Dezembro, 2011
4.1.x	Jelly Bean	16	6,8%	9 de Julho, 2012
4.2.x		17	9,4%	13 de Novembro, 2012
4.3		18	2,7%	24 de Julho, 2012
4.4	KitKat	19	31,6%	31 de Outubro, 2013
5.0	Lollipop	21	15,4%	17 de Outubro, 2014
5.1		22	20,0%	9 de Março, 2015
6.0	Marshmallow	23	10,1%	5 de Outubro, 2015

Tabela 3.1: Versões *Android*

da aplicação:

- Acesso à *internet*, possuir um constante acesso à *internet*, no sentido de conseguir efetuar todas as atualizações necessárias ao servidor, para que seja possível acompanhar o estado do serviço em tempo real.
- Câmara fotográfica, com o objetivo de que o utilizador seja capaz de capturar em fotografia algum momento ou situação que considere ser de alguma relevância para o serviço, tendo então a possibilidade de anexar a mesma ao processo.
- Global Positioning System ([GPS](#)), com o objetivo de se pode registar através do sistema de geolocalização, as coordenadas dos pontos associados ao início e fim de cada fase do serviço.

3.2 Funcionalidades

As principais funcionalidades da aplicação são as seguintes:

- **Obter serviços** - posteriormente à criação e atribuição de um serviço ao motorista por parte do gestor de serviços, o camionista poderá receber na aplicação, informação relativa ao respetivo serviço a realizar como, por exemplo, descrição do serviço, fases a desempenhar, cliente, moradas, entre outros.

- **Registrar informação das fases** – durante o decorrer do serviço, o utilizador terá que ir anotando determinadas informações como: o peso da carga, quilómetros do veículo, horas, e pontos de coordenadas relativamente ao início e fim de cada uma das fases do serviço.
- **Registrar despesas** – existe a possibilidade de registar todas as despesas obtidas no desenrolar do serviço como, por exemplo, despesas com gasóleo, *adblue*, portagens, entre outros.
- **Anexar fotografias** – como já mencionado anteriormente, o camionista pode tirar fotografias e anexá-las ao serviço através da aplicação.
- **Criar uma guia de transporte** – é possível apontar tanto o tipo de mercadoria que transporta, como o seu volume, natureza, e dessa forma criar uma guia de transporte.
- **Histórico** – sempre que exista a necessidade de consultar serviços realizados anteriormente pelo utilizador, este terá ao seu alcance alguma informação relativa a serviços efetuados, como também pode obter todas as guias de transporte criadas pelo mesmo em formato Portable Document Format ([PDF](#)).

Em comparação com as aplicações analisadas na secção anterior, a Transportes App consegue-se destacar pela possibilidade de anexar fotografias ao serviço e pela criação de guias de transporte mencionando as mercadorias que transporta. À semelhança da Wtransnet somente está disponível para dispositivos *Android* e da mesma forma que a LSNet a aplicação consegue reduzir e simplificar de alguma forma o trabalho administrativo da empresa de transporte de mercadorias. Como ambas as aplicações analisadas, é capaz de tirar partido do sistema de geolocalização e ser bastante simples e intuitiva para o utilizador.

3.3 Arquitetura

A arquitetura escolhida para o desenvolvimento deste sistema, é uma arquitetura cliente-servidor de 3 camadas [27]. Neste tipo de arquitetura temos, fisicamente dois pontos: o cliente entendido como o *smartphone* do camionista e o servidor que corresponde à máquina que poderá estar na empresa ou alojada em algum ponto da *internet* funcionando como serviço *web* que poderá ter todas as bases de dados

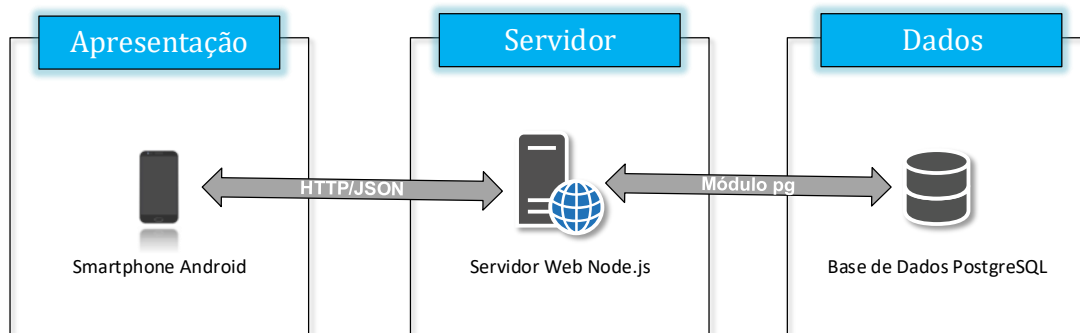


Figura 3.1: Arquitetura de 3 camadas.

e informações relativas aos serviços. A nível do processo lógico funcional, temos 3 camadas que funcionam como 3 módulos distintos (cf. Figura 3.1).

A camada de apresentação é a interface de utilizador, responsável por apresentar informação num formato compreensível ao utilizador. Além de aceitar inputs do utilizador, efetua pedidos e recebe resultados da camada intermédia.

A camada intermédia ou servidor, é onde se encontra grande parte da lógica e código do sistema. Está encarregue de comunicar com a camada de dados, de forma a obter a informação pretendida. Esta informação poderá ainda ser analisada, modificada e posteriormente enviada para a camada de apresentação mostrar ao utilizador.

Por último, temos a camada de dados onde todos os dados são guardados e recebidos da base de dados, e posteriormente reenviados para a camada intermédia para serem analisados.

3.3.1 Ligação Cliente-Servidor

Para a comunicação cliente-servidor, foi desenvolvido uma Application Programming Interface ([API](#)) Representational State Transfer ([REST](#)) no servidor para que fosse possível o *smartphone* efetuar pedidos e receber respostas sempre que necessário para o correto funcionamento da aplicação. Todos os pedidos efetuados pelo cliente (*smartphone*) são executados utilizando o protocolo Hypertext Transfer Protocol ([HTTP](#)), tirando partido dos seus métodos GET e POST. Do outro lado, todas as

respostas enviadas por parte do servidor são objetos JavaScript Object Notation ([JSON](#)), o que facilita bastante a interpretação dos dados recebidos por parte do cliente [36].

3.3.2 Ligação Base de Dados

Toda a comunicação com a base de dados existente está a cargo da [API REST](#) existente no servidor. Deste modo, quando o cliente faz um pedido de informação, introduz ou elimina algum elemento, a *framework* utilizada na construção da [API REST](#) funciona como *middleware* e fica responsável por efetuar todas essas ações na base de dados, e reencaminhar as devidas respostas de volta para o cliente.

3.4 Tecnologias

Para criação de um produto final, foi necessário recorrer a certas tecnologias e ferramentas com o propósito de facilitar e agilizar a idealização, estruturação e, por fim, a elaboração deste projeto.

Para um melhor entendimento do ponto de vista geral do sistema, nesta secção serão apresentadas as tecnologias que foram utilizadas. Desde o sistema operativo utilizado no *smartphone*, passando pela *framework* utilizada para desenvolver a aplicação no servidor que irá receber e enviar os dados para a aplicação móvel. E, por fim, uma breve descrição da base de dados usada para armazenar toda a informação necessária para o correto funcionamento da plataforma.

3.4.1 Android

O sistema operativo *Android* foi criado pela empresa *Android Inc.* Fundada em 2003, o seu foco principal era em câmaras digitais, mas depressa a empresa se apercebeu que este mercado não tinha o tamanho desejado e então redirecionou a sua atenção para os *smartphones*. Em 2005 a *GOOGLE* com uma estratégia de expandir o seu negócio móvel, comprou a *Android Inc.* Cerca de 3 anos depois foi lançado o primeiro dispositivo a utilizar o sistema *Android* [12].

Com o passar dos anos foram surgindo novas versões do seu sistema operativo, cada um com uma designação diferente, mas sempre com nomes alusivos a sobremesas,

Figura 3.2: Logo *Android*.

doces ou bolos (Ex. *Froyo* nome dado a um iogurte gelado, *GingerBread* um doce de origem europeia, *Kitkat* famoso chocolate, entre outros). Apesar de inicialmente ser utilizado em *smartphones*, nos dias de hoje podemos encontrar o sistema *Android* (logo apresentado na Figura 3.2) em diferentes dispositivos, desde *tablets*, *netbook*, *smart tv's*, entre outros. Tornando este o sistema mais utilizado no mercado dos dispositivos moveis [28].

É considerado um sistema de código aberto, o que permite que qualquer programador possa desenvolver aplicações para esta plataforma necessitando somente de um conjunto de ferramentas. Ferramentas estas que estão distribuídas pelas camadas que caracterizam a arquitetura deste sistema [39].

- ***Linux Kernel:***

É a camada mais baixa desta arquitetura, foi construída em cima do núcleo do *Linux*. Através desta é possível serem feitos acessos a componentes de *hardware* e efetuar gestão de processos e memória.

- ***Libraries/Bibliotecas:***

Onde se encontram um conjunto de módulos com características base do *Android*, como *SQLite* (base de dados que guarda dados relativos a aplicação), *WebKit* (motor de navegação web), bibliotecas relativas aos conteúdos multimédia, entre outras.

- ***Android Runtime:***

Mecanismo também pertencente à segunda camada, contém uma componente chave como Java Virtual Machine (**JVM**) apelidada por *Dalvik VM* que permite

compilar e executar as aplicações. E disponibiliza também um conjunto de bibliotecas núcleo para desenvolver as aplicações para o sistema *Android* utilizando a linguagem Java.

- ***Application Framework/Framework* para aplicações:**

Camada que permite aos programadores terem acesso a serviços do sistema através das suas aplicações.

- ***Application* / Aplicação:**

É a camada mais alta de todo sistema, onde se integram todas as aplicações desenvolvidas pelo programador com as aplicações já incluídas com o sistema.

Todo o tipo de ferramentas e documentação necessária para o desenvolvimento deste tipo de aplicações podem ser adquiridas no *website* do fabricante, uma vez que é uma plataforma livre e de consulta pública.

3.4.1.1 Android Studio

Para o desenvolvimento da aplicação para *smartphone* foi utilizado o Integrated Development Environment ([IDE](#)) *Android Studio*, esta escolha deve-se ao facto de este ser o [IDE](#) oficial para desenvolver aplicações *Android*. Baseado no *IntelliJ IDEA*, outro IDE utilizado para desenvolver aplicações *Java*, o *Android Studio* consegue oferecer todas as ferramentas necessárias para potenciar a construção de aplicações *Android*, como por exemplo: *Android Software Development Kit* ([SDK](#)), uma plataforma de ferramentas *Android*, um emulador rápido e rico em recursos, uma imagem do sistema, integração com *GitHub*, entre outros. Sendo a sua primeira versão lançada em Maio de 2013 (v0.1.x), este [IDE](#) no passar dos anos foi evoluindo, não só em termos de performance e correção de erros mas também com a introdução de novas ferramentas para facilitar a vida aos programadores, encontrando-se atualmente na versão 2.1.2 (lançada em Junho de 2016) [[11](#)].

3.4.1.2 Desenvolvimento de aplicações Android

Os principais componentes utilizados no desenvolvimento da aplicação *Android* são:

- ***Activities***

É o componente com maior importância numa aplicação *Android*, fornece o ecrã

onde será apresentado a interface de utilizador e contém toda a lógica funcional da aplicação (código) [20]. Uma aplicação tipicamente tem varias *activities*, sendo uma delas a principal (*main*) que é executada no início da aplicação. Cada uma das *activities* é capaz de abrir outra, podendo cada uma delas desempenhar uma ação diferente [14].

- ***Intents***

São objetos de mensagem usados para comunicar entre várias partes do sistema *Android* e para iniciar *Activities* [16].

- ***Widgets***

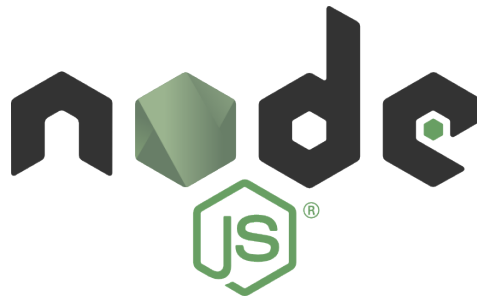
Elementos (principalmente visuais) da interface do utilizador, que podem ser criados, personalizados e usados no ecrã da aplicação. Os *widgets* mais utilizados na aplicação são:

- *TextView*
- *EditText*
- *Button*
- *ImageButton*
- *RelativeLayout*
- *GridLayout*

- ***AsyncTask***

Permite que seja possível executar múltiplas ações em *background* e que os seus resultados sejam publicados no processo (*thread*) principal de forma a transmitir informação ao utilizador, sem que seja necessário gerir processos e *handlers*. O seu funcionamento é composto por 3 etapas [15]

- *onPreExecute()* corre na *thread* principal antes da tarefa começar, usado para alguma configuração ou até para informar o utilizador que está prestes executar a tarefa.
- *doInBackground()* método onde é colocado todo o código que a ser executado em *background*. Uma vez finalizada a execução, o resultado é enviado para o *onPostExecute*.
- *onPostExecute()* é executado imediatamente a seguir ao *doInBackground*, e é onde se atualiza a interface gráfica consoante os resultados recebidos.

Figura 3.3: Logo *Node.js*. [29]

- **Fragments**

Como o nome indica, são fragmentos de uma *activity*. Podendo mesmo ser visto como *sub-activities*, em que uma *activity* poderá combinar e mostrar um ou múltiplos fragmentos no ecrã ao mesmo tempo. Possuem o seu próprio ciclo de vida, eventos de entrada e são adicionadas e removidas durante o correr de uma *activity* [13].

3.4.2 Node.js

Tendo sido lançada em 2009 pelo Ryan Dahl, *Node.js* é uma plataforma de código aberto para desenvolvimento de aplicações *web* do lado do servidor (logo na Figura 3.3). Com grande parte dos seus módulos escritos em *JavaScript*, esta plataforma tem vindo a crescer e a ganhar bastante popularidade no mundo da programação *web* [30]. “With over 3.5 million users and an annual growth rate of 100 percent, *Node.js* is emerging as a universal platform used for web applications, IoT, and enterprise.” [29]

Com módulos que permitem fazer gestão de sistema de ficheiros, de redes (Domain Name System ([DNS](#)), [HTTP](#), Transmission Control Protocol ([TCP](#)), User Datagram Protocol ([UDP](#))), *buffers*, entre muitas outras funções. Contém também uma [API](#) especialmente desenhada para reduzir ao máximo a complexidade existente na programação de aplicação de servidor.

Esta *framework* foi escolhida para o desenvolvimento da [API REST](#) deste projeto, por existir já alguma familiaridade com a plataforma, e por a ser uma plataforma muito utilizada, com muita documentação e imensos módulos disponibilizados.

3.4.2.1 Módulos

Os módulos utilizados neste projeto foram:

- **HTTP:** Módulo usado para a criação de um servidor [HTTP](#). [32]
- **Express:** Módulo simples e flexível que providencia algumas ferramentas para servidores [HTTP](#) [4]
- **Body-parser:** Possibilita fazer o *parse* do corpo dentro dos pedidos *HTTP*, utilizando a propriedade *req.body*. [3]
- **Morgan:** Módulo que permite a criação de um *log* de eventos [7]
- **Multiparty:** Módulo que permite a receção de ficheiros fazendo o *parse* dos pedidos de tipo “*multipart/form-data*”. [8]
- **Fs:** Módulo que disponibiliza funções síncronas e assíncronas de *input/output* do sistema de ficheiros. [5]
- **Pg:** Módulo que disponibiliza funções de cliente *PostgreSQL* para o *Node.js* [10]
- **Jsonwebtoken:** Módulo que providencia métodos para tratamento de *tokens* do cabeçalho *HTTP*. [6]
- **Html-pdf:** Módulo utilizado para converter código HyperText Markup Language ([HTML](#)) em um ficheiro [PDF](#).

3.4.2.2 WebStorm

Para efetuar o desenvolvimento em *Node.js* foi utilizado o *WebStorm*, um [IDE](#) de *Javascript* criado pela equipa *JetBrains*, os mesmos criadores do *IntelliJ IDE* já mencionado. *WebStorm* é um [IDE](#) bastante *user friendly*, com uma interface muito intuitiva, que faz a análise do projeto de forma a poder oferecer as melhores sugestões de código, fornece varias ferramentas de *debug*, trabalha com diversos sistemas de controlo de versões, entre outras características que tornam este [IDE](#) uma boa escolha [25]. A contrapartida deste *software* é ser pago, tendo sido utilizada no decorrer deste projeto uma licença de estudante.

3.4.3 PostgreSQL

O *PostgreSQL* (logo na Figura 3.4) é um poderoso Sistema de Gestão de Bases de Dados Objeto-Relacional (SGBDOR) de código aberto, desenvolvido pela *PostgreSQL Global Group* [22]. Tendo a sua primeira versão sido lançada em 1996, esta base de dados conta já com mais de 20 anos de desenvolvimento, fazendo-se notar na grande reputação em pontos como *performance*, confiabilidade, integridade de dados, segurança e por ser multiplataforma, sendo compatível com sistemas *Windows*, *Mac OS* e *Linux* [23]. Inclui a maior parte dos tipos de dados do *SQL 2008*, suporta o armazenamento de objetos binários grandes, incluindo vídeo, imagens e sons. Existe também uma lista de extensões e recursos avançados que podem ser usados com o propósito de aumentar o número de funcionalidades existentes, como por exemplo a utilização do *PostGis* que permite com que o *PostgreSQL* aceite objetos geográficos e permita que seja usado como uma base de dados espacial para um Sistema de Informação Geográfica (SIG) ou Geographic Information System (GIS) [22].

A escolha desta base de dados para o desenvolvimento do projeto deveu-se ao facto da empresa *D&C Software* utilizar *PostgreSQL* em grande parte dos seus projetos, existindo mesmo tabelas já criadas que serão utilizadas por este projeto.



Figura 3.4: Logo *PostgresSql.js* [40].

3.4.3.1 pgAdmin

Para criação e manutenção da base de dados foi utilizado o *pgAdmin*. É uma plataforma *Open Source* de administração e desenvolvimento *PostgreSQL*, pode ser usada em sistemas *Linux*, *Windows*, *Mac OSX* entre outros e suporta a versão 7.3 do *PostgreSQL* e superior. Desenvolvido por uma comunidade de especialistas em *Postgres* de todo mundo, este *software* contém um ambiente gráfico que torna bastante simples ao utilizador a administração das base de dados, permitindo desde a criação de simples *queries* ao desenvolvimento de complexas base de dados [31].

Capítulo 4

Implementação do Sistema

Neste capítulo é feita uma análise mais detalhada sobre a implementação do sistema, começando pela base de dados e as suas tabelas, passando pelas rotas e algumas funções implementadas no servidor acabando numa abordagem detalhada por alguns ecrãs importantes da aplicação do *smartphone*.

4.1 Base de Dados

Para um correto funcionamento da aplicação, é necessário que exista um local onde se possa armazenar e consultar informação sempre que necessário. Para que isso aconteça recorreu-se à criação de uma base de dados tanto no lado do servidor como no lado do cliente (*smartphone*).

4.1.1 Servidor

Como já mencionado no capítulo anterior, a base de dados utilizada foi construída em PostgreSQL. Depois de feito um levantamento de toda a informação que seria necessária para que houvesse um bom funcionamento da aplicação, chegou-se a conclusão que seria necessário a construção de 8 tabelas. Estas tabelas guardam dados relativos aos clientes, motoristas, veículos que foram utilizados, serviços e as suas várias fases, despesas contraídas no decorrer dos serviços, fotos e guias de transportes.

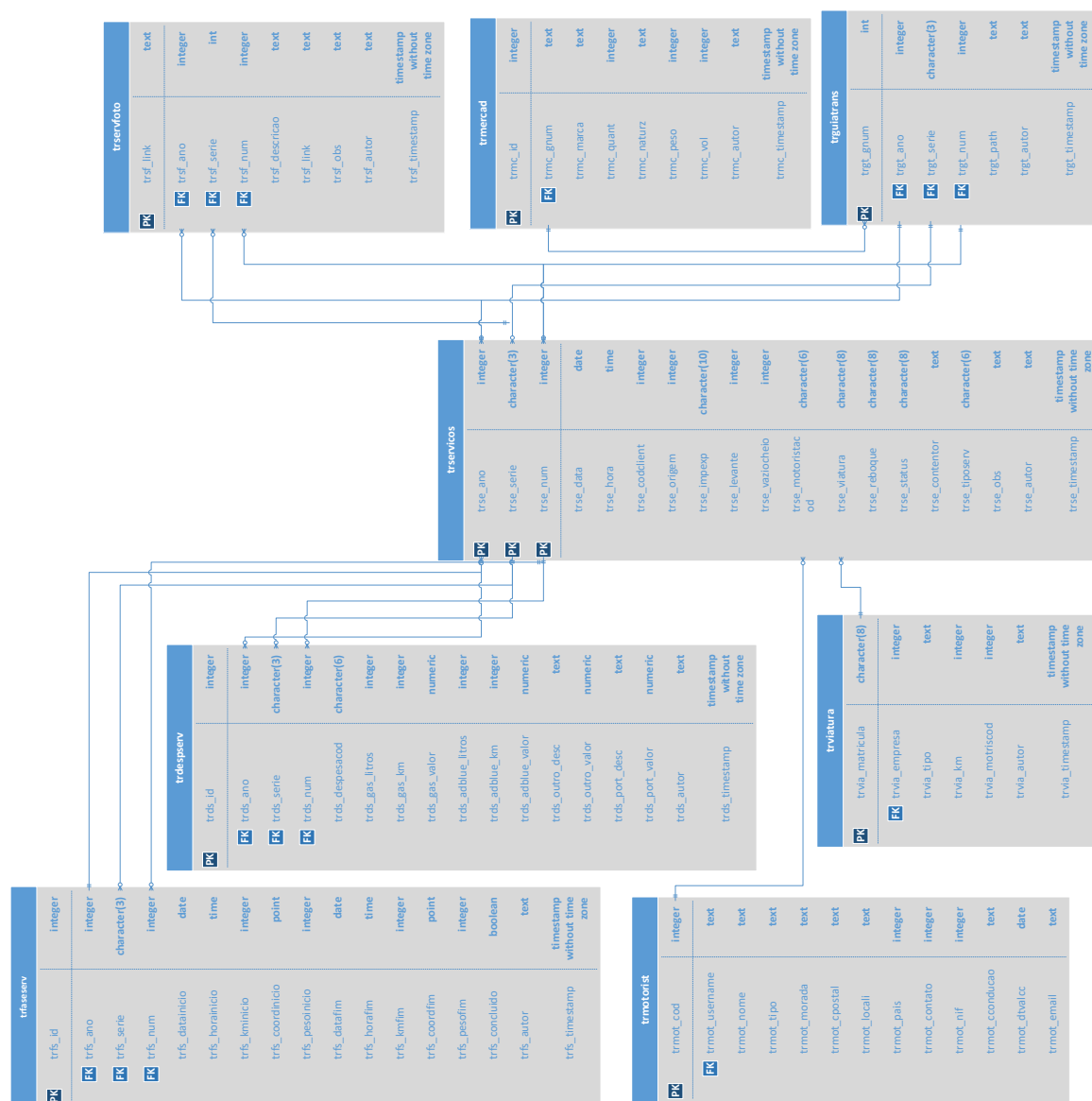


Figura 4.1: Base de Dados Servidor.

Durante toda a idealização de construção da base de dados, houve a preocupação em tentar manter toda a estrutura o mais simples e otimizada possível, de maneira a evitar a necessidade de criação de *queries* muito complexas que poderiam de certa forma agravar a performance do sistema. Posto isto, é de realçar que todas as tabelas criadas são de extrema importância para que se possa tirar partido de todas as funcionalidades da aplicação.

A Figura 4.1 mostra a base de dados criada.

4.1.2 Cliente

Com o intuito de tentar tornar a aplicação de certa forma funcional em períodos em que a rede de *internet* esteja indisponível, e também para criar alguma redundância de informação e melhoramento de performance, quando executada pela primeira vez, a aplicação cria uma base de dados SQLite no *smartphone* do motorista.

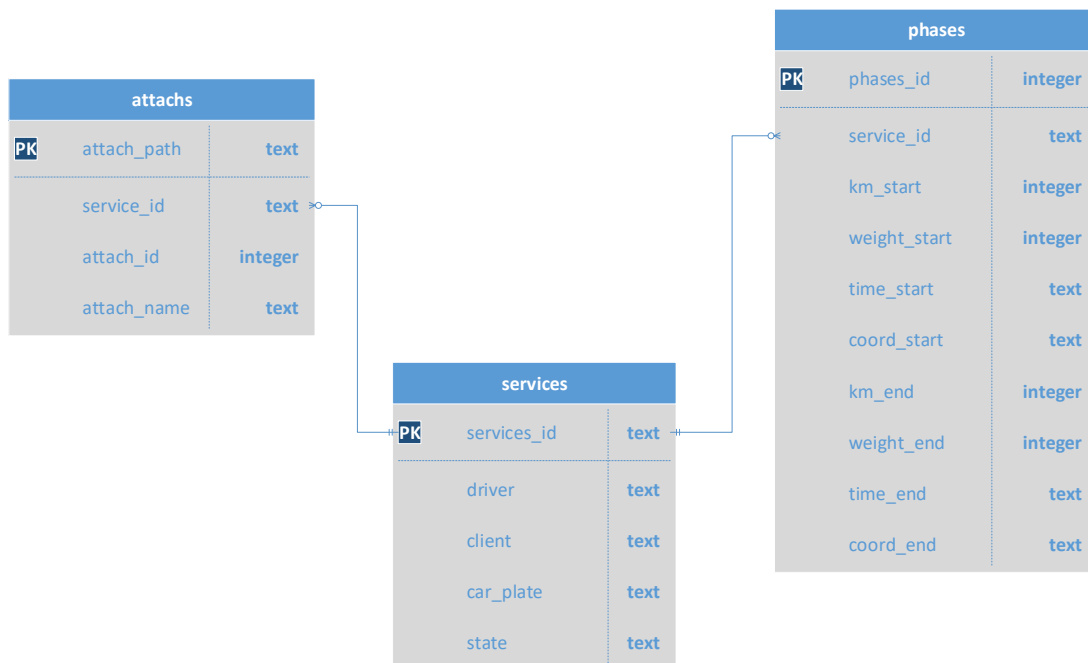


Figura 4.2: Base de Dados Cliente.

Esta base de dados é bastante mais pequena que a base de dados criada no servidor,

pois contém somente tabelas que permitem o armazenamento de informação que se entende como mais importante a ser preservada em caso de falta de conectividade. Estamos a falar de dados como informação do motorista, do veículo utilizado, anexos (fotografias) e de informação relativa as fases do serviço (registo de quilómetros, peso de carga, horas e coordenadas).

A base de dados criada, é apresentada na Figura 4.2.

4.2 API REST

Conforme descrito no capítulo anterior, o sistema contém uma [REST API](#) construída em Node.js alojada num domínio na *internet* e terá como função de receber e responder a pedidos e funcionar como intermediário entre a base de dados e a aplicação de *smartphone*.

Nesta secção são detalhadas algumas das funções desempenhadas por esta [API](#), rotas e tipos de comunicações efetuadas.

4.2.1 Rotas

No decorrer do desenvolvimento desta [REST API](#) foram definidas algumas rotas que são utilizadas para que a aplicação possa solicitar informação ou até mesmo registar dados referentes aos serviços. Cada uma destas rotas é um Uniform Resource Locator ([URL](#)) com alguns parâmetros associados e um método [HTTP](#) específico: GET ou POST(cf. Tabela 4.1).

O propósito de estabelecer várias rotas, foi para que fosse possível estabelecer uma ação para cada uma delas e, dessa maneira, tornar mais fácil à [REST API](#) saber que parâmetros receber e que operações a desempenhar para cada rota em particular.

Rota	Método	Função
/transportes/login	POST	Verifica se o nome de utilizador e palavra passe recebidos existe e estão corretos.
		Caso as credenciais estejam corretas, é gerado e adicionado um token ao cabeçalho http.

/transportes/confirmInfo	POST	Envia um conjunto de informação para a aplicação com o objetivo do utilizador confirmar se estão corretos.
/transportes/sendConfirmInfo	POST	Modifica na base de dados os campos que foram corrigidos pelo utilizador.
/transportes/addVehicle	POST	Adiciona um carro à base de dados.
/transportes/description	GET	Envia a descrição do serviço que esteja a decorrer. Caso não esteja nenhum serviço a decorrer envia a descrição do que esteja pendente, associado ao utilizador.
/transportes/updateContainer	POST	Atualiza na base de dados a matrícula do contentor associado ao serviço.
/transportes/currServ	POST	Envia para a aplicação toda a informação referente ao serviço atribuído ao utilizador.
/transportes/updatePhase	POST	Regista na base de dados os dados referentes a uma fase do serviço. Hora, quilómetros, peso e coordenadas associadas ao início e fim da respetiva fase.
/transportes/servicesHistory	GET	Envia uma pequena descrição de todos os serviços desempenhados pelo utilizador.
/transportes/phaseHistory	POST	Envia os dados associados as fases de um determinado serviço que o motorista tenha efetuado.
/transportes/uploadPhoto	POST	Recebe uma fotografia enviada pela aplicação.
		Cria (caso não exista) uma pasta associada ao serviço, onde ficarão guardadas todas as fotos recebidas.
		Regista na base de dados o nome e o link para a pasta onde ficou armazenada a foto.
/transportes/getPhotosNames	GET	Envia o nome de todas as fotos associadas a determinado serviço.
/transportes/getPhoto	POST	Envia para a aplicação a fotografia com o nome pedido.

/transportes/getExpenses	GET	Envia todas as despesas de determinado serviço
/transportes/addExpenses	POST	Adiciona na base de dados alguma despesa efetuada durante o serviço.
/transportes/gTransp	GET	Envia para o smartphone todas as informações relevantes a construção da guia de transporte de certo serviço.
/transportes/saveGuideTransp	POST	Recebe e regista todas as mercadorias na base de dados.
		Cria uma guia de transporte em formato PDF com todas as mercadorias recebidas, e guarda na pasta das guias de transporte.
/.../addItemGuideTransp	POST	Adiciona uma mercadoria a base de dados.
/.../deleteItemGuideTransp	POST	Elimina alguma mercadoria da base de dados.
/transportes/gTranspHistory	GET	Envia o nome de todas as guias de transporte existentes, criadas pelo utilizador.
/transportes/OpenGTransp	POST	Envia o ficheiro PDF de uma guia de transporte para que possa ser visualizado na aplicação.
/transportes/endService	POST	Altera o estado de um serviço na base de dados para encerrado.
/transportes/logout	GET	Apaga todas as variáveis temporárias associadas ao utilizador.

Tabela 4.1: Tabela de rotas

4.2.2 Tokens

O protocolo [HTTP](#) é considerado *stateless*, isto é, não guardar informação sobre cada ligação, interpretando dessa forma cada pedido como um pedido independente. Por esta razão, surgiu a necessidade de utilizar um mecanismo que permitisse autenticar o utilizador em cada ligação que fosse estabelecida, sem a utilização de credenciais de acesso. O método escolhido foi a utilização de *tokens*, por ser um mecanismo com pouco impacto a nível de performance e que acrescenta algum tipo de segurança nas comunicações.

O modo de funcionamento é extremamente simples, quando o utilizador se autenticar na aplicação, é feita uma ligação ao servidor com a finalidade de verificar se as credenciais introduzidas estão corretas, caso estejam, é gerado um *token* que é anexado ao cabeçalho [HTTP](#) e enviado para o cliente. O cliente do seu lado trata de guardar o *token* recebido e utiliza-lo nas ligações seguintes, de forma a se autenticar perante o servidor. Cada um destes *token* expira após um certo tempo definido, obrigando dessa forma o utilizador a voltar a fazer *login* na aplicação.

A fim de criar esta técnica de autenticação na *framework* Node.js, foi necessário recorrer ao uso do módulo *Jsonwebtoken* para a criação e verificação dos *tokens*. Para gerar um *token*, após a verificação das credenciais do utilizador, é utilizada a função *jwt.sign* que recebe como argumentos, uma string, uma palavra secreta e um campo para opções como: o algoritmo a utilizar (sendo HS256 por defeito), tempo para o *token* expirar, alguma descrição, entre outros. Para verificar se um *token* é válido é utilizada a função *jwt.verify*, que tem como argumentos o *token*, a palavra secreta utilizada anteriormente e um *callback* onde é verificado se ocorreu erro ou se o *token* é válido.

As rotas mostradas na Tabela 4.1 à exceção da “/transportes/login”, todas passam pela função de verificação do *token* antes de executarem as suas ações. A rota de *login* passa pela função de criação do *token*.

4.2.3 Ligação à Base de Dados

Com a [REST API](#) a funcionar como intermediário entre a aplicação do *smartphone* e a base de dados, é necessário que esta consiga estabelecer ligação à base de dados e efetuar *queries*. Para que isso aconteça é utilizado o módulo *pg* do Node.js.

Com este módulo é possível configurar e criar um *pool* de clientes, e dessa forma possibilitar a existência de várias ligações em simultâneo à base de dados. Evita-se a criação de uma ligação cliente sempre que seja feito um pedido, podendo ser feita a reutilização da conexão da *pool*.

4.2.4 Guia de Transporte

Para que fosse possível a elaboração de um ficheiro [PDF](#) com os dados de uma Guia de Transporte procedeu-se a instalação do módulo *Html-Pdf*. Utilizando a função

`pdf.create().toFile()` é possível dar como argumento uma *string* contendo uma pagina no formato [HTML](#) ao método `create()`, ao método `toFile()` é dado uma *string* com o destino onde o ficheiro vai ser guardado e uma *callback*.

4.2.5 Fotografias

No envio de fotografias usando o protocolo [HTTP](#), os pedidos utilizados são do tipo “*multipart/form-data*”, o que faz com que cada ficheiro seja enviado em fragmentos, sendo da responsabilidade do Node.js receber todos estes pedaços para obter o ficheiro original. Para que fosse possível este tipo de funcionamento por parte do Node.js, foi instalado o módulo *Multipart*, que permite a criação de um objeto `multipart.Form()` e então o uso do método `form.parse()` que tem como argumento um *request* recebido, fazendo com que a função `form.on()` emita vários eventos. Quando esta função é usada com a *string* “*error*”, podemos verificar a ocorrência de alguns erros, se o valor da *string* dada for “*file*” pode-se verificar que ficheiro foi recebido e por último com o valor da *string* sendo “*close*” podemos efetuar alguma ação pois o *parse* dos fragmentos foi concluído.

4.3 Aplicação Android

Neste secção é descrita de uma forma mais pormenorizada a aplicação de *smartphone* utilizada pelos motoristas. São abordados pontos como a interface gráfica, as funcionalidades implementadas e o modo de funcionamento da respetiva aplicação.

4.3.1 Login

O *login* é o primeiro ecrã que o utilizador presencia ao abrir a aplicação. Neste ecrã é pedido que o utilizador introduza as suas credenciais de acesso à aplicação (nome de utilizador e palavra passe) como se verifica na Figura [4.3](#).

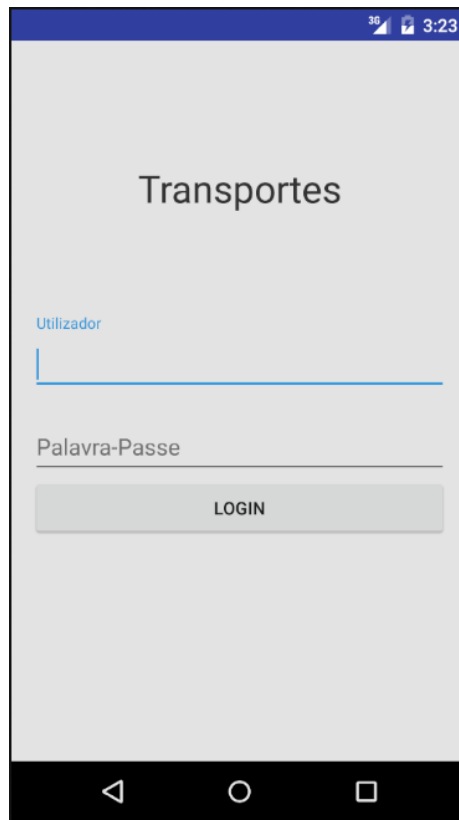


Figura 4.3: Transportes App - Login

Ao ser premido o botão de *login*, a aplicação realiza uma conexão com o servidor com o intuito de verificar se é possível estabelecer comunicação. Caso o *feedback* seja positivo, então é criada uma nova *thread* para realizar o pedido de *login*. Os dados introduzidos pelo utilizador são enviados através do método POST do protocolo [HTTP](#) e a resposta recebida, é um objeto [JSON](#), contendo um campo “*message*” onde se pode verificar se o *login* foi efetuado corretamente.

4.3.2 Descrição do Serviço

Após passar por um ecrã de confirmação de dados como: matrícula do veículo e do reboque e número de quilómetros do veículo, deparamo-nos com o ecrã de descrição do serviço (presente na Figura 4.4). Esta descrição poderá ser do serviço que esteja a decorrer, caso não haja nenhum a decorrer, será a descrição do próximo serviço a ser efetuado pelo motorista.

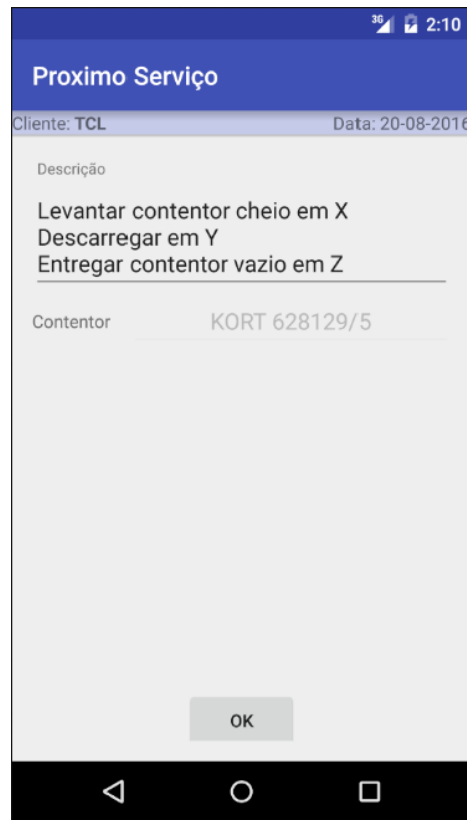
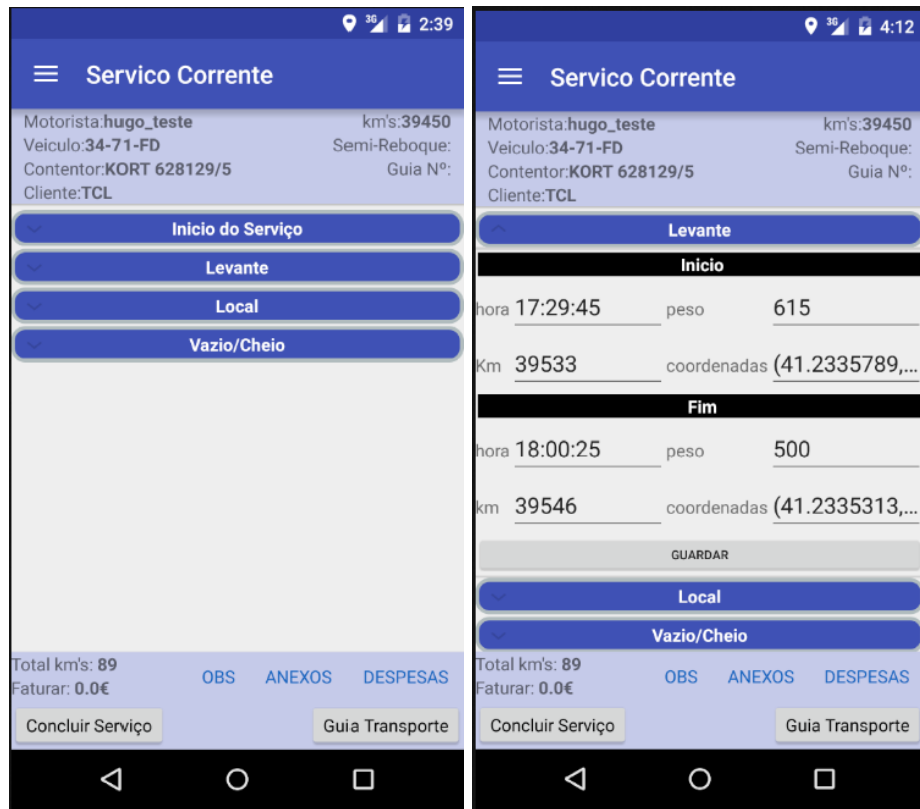


Figura 4.4: Transportes App - Descrição do serviço

O campo referente à descrição é editável, pelo que o utilizador tem a possibilidade de modificar a sua descrição em caso de alguma alteração no serviço, existe também uma caixa de texto destinada à introdução da matrícula do contentor, que é de preenchimento obrigatório e assim que esteja definida não poderá ser alterada. Podemos obter informação do nome do cliente que requisitou o serviço e a data a efetuar o mesmo. Todos os campos são enviados através do método POST e os dados recebidos através de um objeto [JSON](#).

4.3.3 Serviço

Este é o ecrã principal da aplicação (cf. Figura 4.5), onde encontramos um cabeçalho com informação referente ao camionista, veículo e serviço. No rodapé existem botões de acesso a páginas que podem ser importantes no desenrolar do serviço. E por fim, no centro encontramos uma lista (*expendableList*) com as várias fases pertencentes ao serviço, em que ao carregarmos em uma delas será mostrado os campos para introdução das horas, pesos, quilómetros e coordenadas de início e fim de cada uma das fases.



(a) Ecrã de menu.

(b) Disponibilizar camiãõ.

Figura 4.5: Transportes App - Serviço a decorrer

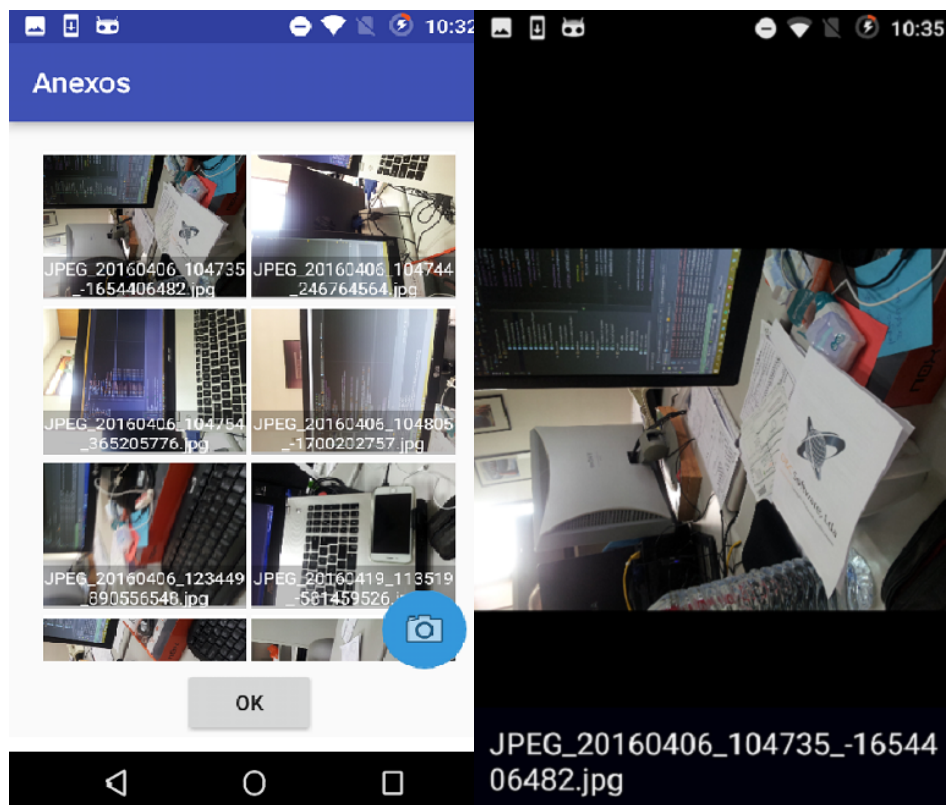
Os campos de hora e das coordenadas, são preenchidos automaticamente quando selecionados. Sendo possível também no caso de as horas serem introduzidas manualmente, existindo uma máscara de formatação e uns alertas associados à caixa de texto de forma a facilitar e garantir que sejam introduzidos valores válidos. Da mesma forma, foram definidos alertas para garantir que não sejam deixados campos vazios, introduzidos valores na secção de fim da fase sem que sejam introduzidos os respetivos valores de início e impedir que sejam introduzidos dados numa fase, sem que a anterior esteja completa.

Assim que seja carregado o botão “guardar” existente numa das fases, todos os dados da respetiva fase introduzidos pelo utilizador, serão enviados para o servidor, para que este atualize os mesmos na base de dados. De forma de criar alguma redundância e solucionar o problema de falta de rede que poderá acontecer durante o decorrer de um serviço, as informações registadas são armazenadas na base de dados local do *smartphone*.

Quando o serviço estiver terminado, o motorista poderá carregar no botão de “Concluir Serviço”, então o sistema irá verificar se as fases estão todas completas, enviar para o servidor informação que o serviço foi completo, e na aplicação voltar para a página de descrição, onde será mostrada informação de um novo serviço.

4.3.4 Anexos

Há situações em que pode ser necessário tirar uma fotografia de forma de captar alguma situação que o motorista ache relevante associar ao serviço, como por exemplo, carga danificada, danos feitos ao veículo ou atrelado, entre outros. Estas imagens podem ser capturadas pela aplicação recorrendo ao uso da câmara fotográfica do *smartphone* e mostradas num aspeto de grelha (utilizando uma *GridView*, como se verifica na Figura 4.6a), em que é possível a visualização de cada uma das fotos em ecrã completo (cf. Figura 4.6b).



(a) Grelha de fotos.

(b) Foto em *fullscreen*.

Figura 4.6: Transportes App - Anexos

Com o objetivo de otimizar a velocidade e reduzir a utilização de *internet*, todas as fotos tiradas além de serem enviadas para o servidor, são armazenadas localmente no *smartphone*. Tendo somente de recorrer ao *download* das fotos que não tenham sido tiradas do mesmo *smartphone*.

4.3.5 Despesas

Como forma de registrar todas as despesas associadas ao serviço em curso, existe o ecrã de despesas (cf. Figura 4.7). Neste ecrã o motorista pode apontar desde valor de portagens, a gasóleo, *adblue* ou até outro tipo de despesa que não esteja mencionada na aplicação, indicando uma descrição e o valor respetivo.



Figura 4.7: Transportes App - Despesas

4.3.6 Guia de Transporte

Sempre que seja necessário criar uma guia de transporte contendo as mercadorias a transportar, a aplicação poderá facilitar esse processo. Através do ecrã de guia de transporte (cf. Figura 4.8), é possível adicionar informação relativa às mercadorias, como, quantidade, natureza, peso e volume. Ao carregar no botão de “guardar” toda informação é enviada para o servidor, que além de ter a tarefa de adicionar às mercadorias na base de dados, tem a função de criar um ficheiro PDF com a mesma informação.

Guia de Transporte

Expedidor: Rogério Leal & Filhos Sa
 Destinatário: REPNUMMAR
 Local de carga: R. Padre Manuel da Nóbrega
 Local de descarga: R. da Boa Viagem 114

Numeros/ Marcas	Quantidade	Natureza	Peso	Volume
Sapatos	36	calcado	39	15
Calças	36	vestuario	16	34

+

Valor a Pagar	Expedidor	Destinatario
Transporte		
Sobretaxa		
Reexpedição		
Total		
IVA %		
Total + IVA		

Reembolso

Motorista: hugo_teste
 Guia Nº: GT2 Data: 2016-03-29

GUARDAR

Figura 4.8: Transportes App - Criar Guia de transporte

4.3.7 Barra de menu

Para ter acesso a outras funcionalidades que não estão diretamente ligadas com o serviço a decorrer, foi criado um menu lateral deslizante. Neste menu (cf. Figura 4.9), está presente a opção de consultar um histórico dos serviços já efetuados pelo utilizador, as guias de transportes criadas pelo mesmo, terminar a sessão ou até voltar

para o ecrã do serviço a decorrer.

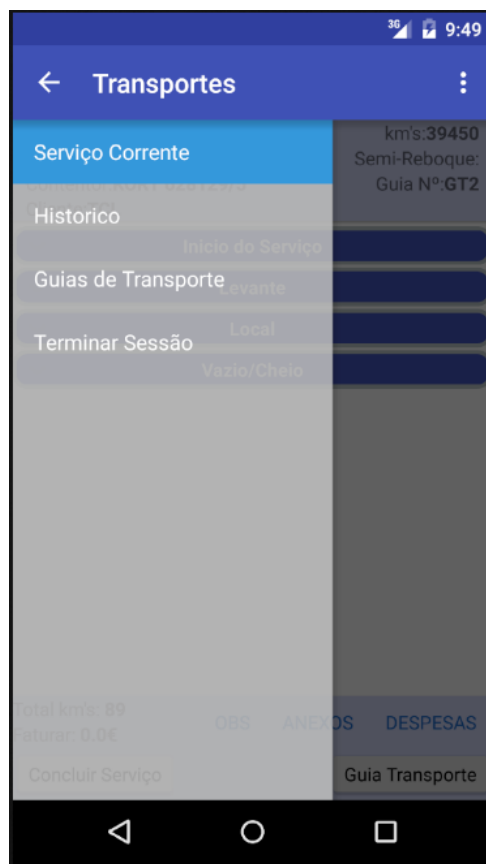
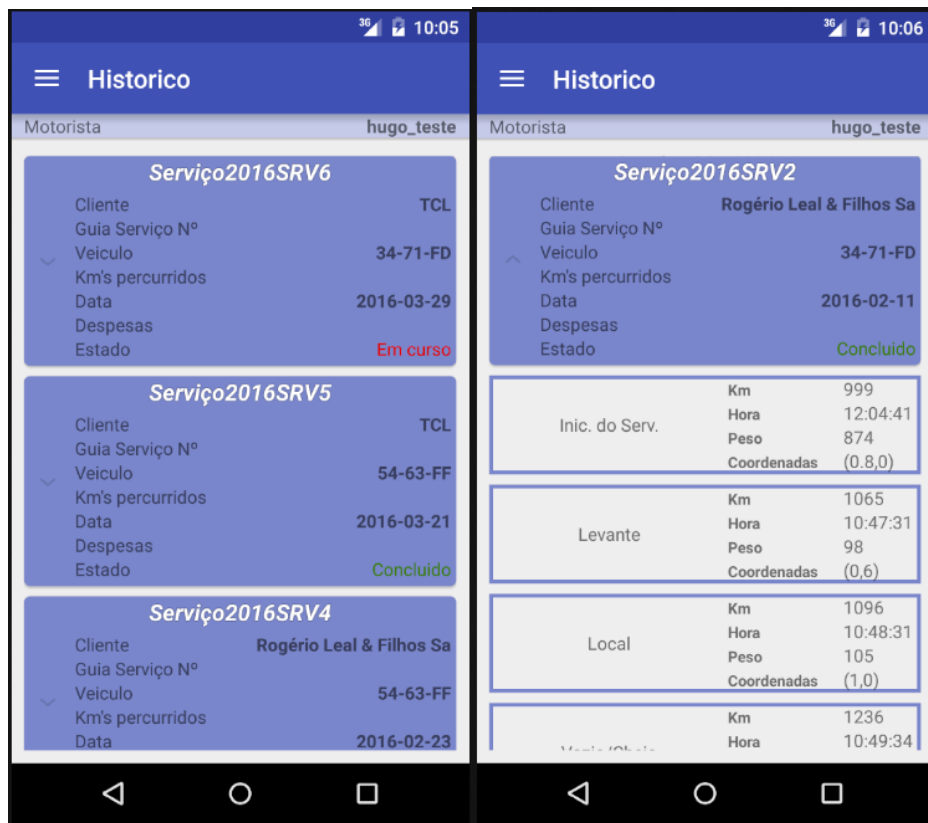


Figura 4.9: Transportes App - Barra de menu

Este tipo de menu é a forma mais fácil, organizada e intuitiva de disponibilizar mais funcionalidades ao utilizador sem ter que sobrecarregar a página principal com botões ou hiperligações.

4.3.8 Histórico

Como mencionado anteriormente, nesta página é possível consultar um resumo de todos os serviços desempenhados pelo utilizador. Em cada um dos serviços vem mencionado o cliente, o veículo, o estado de serviço (em curso ou concluído), a data, entre outros (cf. Figura 4.10a). Para que seja possível visualizar informação relativa às fases de um determinado serviço, basta carregar no mesmo e serão exibidas todas as fases e os seus registos relativos ao fim de cada uma das respetivas fases (cf. Figura 4.10b).



(a) Serviços realizados

(b) Fases de serviço realizado

Figura 4.10: Transportes App - Histórico de serviços

4.3.9 Visualizar guias de Transporte

Este ecrã serve para visualizar todas as guias de transporte criadas pelo utilizador, em formato PDF. Ao abrir a página, a aplicação faz um pedido ao servidor, pelo nome de todas as guias criadas pelo motorista e mostra utilizando um *ListView* (cf. Figura 4.11a). Assim que uma guia seja selecionada, é imediatamente descarregado o ficheiro correspondente, e aberto através de uma aplicação que esteja instalada no *smartphone* para visualizar ficheiros PDF (cf. Figura 4.11b). Caso não exista nenhuma aplicação definida para abrir PDF será mostrada uma mensagem de erro.

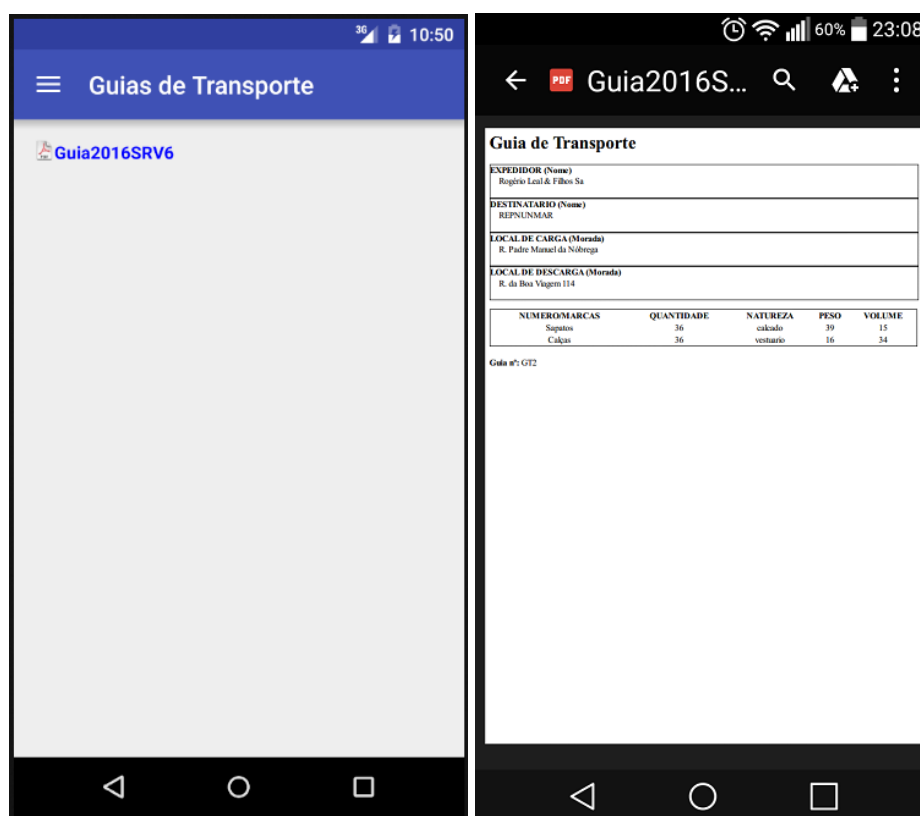


Figura 4.11: Transportes App - Guias de Transporte

Capítulo 5

Conclusão

Torna-se cada vez mais comum as empresas aderirem às novas tecnologias de forma a poderem melhorar os seus desempenhos, e oferecerem um melhor serviço aos seus clientes. É nesse sentido que a Transportes App foi criada, agora as empresas de transportes de mercadorias poderão usufruir de uma ferramenta que lhes permite otimizar os seus processos de gestão de serviços.

5.1 Contribuições e satisfação dos objetivos

Este projeto iniciou com um conjunto de objetivos e etapas idealizadas para a correta criação da plataforma, que passavam pela criação de uma aplicação *Android* para *smartphones*, de uma aplicação servidor desenvolvida em *Node.js* e desenho e criação de uma base de dados *PostgreSQL*. No seu decorrer, foram encontradas algumas dificuldades, que foram ultrapassadas com ajuda e acompanhamento do grupo de trabalho, que tornou este projeto bastante enriquecedor tanto a nível de aprendizagem, visto a grande diversidade de tecnologias abordadas, como também a nível profissional, por ter integrado pela primeira vez um projeto em ambiente empresarial.

No final deste projeto, é possível verificar que foram cumpridos todos os objetivos propostos, tendo sido implementadas todas as funcionalidades inicialmente estipuladas.

5.2 Dificuldades encontradas

As principais dificuldades encontradas no decorrer do trabalho, foram na iniciação à tecnologia *Android*, visto a ser uma tecnologia com a qual nunca tinha trabalhado o que me levou a ter que fazer um estudo para aprender a trabalhar com o sistema, tendo mesmo que recorrer a leitura de documentação durante o desenvolvimento do projeto.

Outra dificuldade encontrada foi na integração/comunicação entre os sistemas: *Android*, *Node.js* e base de dados *PostgreSQL*. Inicialmente, no envio de dados simples, e posteriormente no envio de ficheiros como: fotos e documentos do formato [PDF](#).

5.3 Trabalho futuro

Apesar de terem sido cumpridos os objetivos, existem sempre aspetos que podem ser melhorados tendo em vista um maior número de funcionalidades e melhor desempenho da aplicação.

Marcar reuniões com possíveis clientes (empresas de transporte de mercadorias), com o objetivo de reunir novas funcionalidades que possam ser implementadas na aplicação, podendo mesmo ajustar a Transportes App às suas necessidades.

Melhorar performance e a interface gráfica da aplicação, tornando-a ainda mais rápida, simples e apelativa aos seu utilizadores.

Por ultimo, como foi inicialmente estipulado, a aplicação foi desenvolvida para sistemas *Android*, pelo que poderá passar por um trabalho futuro o desenvolvimento da aplicação para sistemas *IOS* e *Windows Phone*.

Apêndice A

Acrónimos

ATRAM	Associação Nacional de Transportadores Públicos	SDK	Software Development Kit
GPS	Global Positioning System	IDE	Integrated Development Environment
PDF	Portable Document Format	DNS	Domain Name System
API	Application Programming Interface	TCP	Transmission Control Protocol
REST	Representational State Transfer	UDP	User Datagram Protocol
HTTP	Hypertext Transfer Protocol	SGBDOR	Sistema de Gestão de Bases de Dados Objeto-Relacional
HTML	HyperText Markup Language	SIG	Sistema de Informação Geográfica
JSON	JavaScript Object Notation	GIS	Geographic Information System
JVM	Java Virtual Machine	URL	Uniform Resource Locator
IDE	Integrated Development Environment		

Referências

- [1] Aplicação lsnet. <http://www.luis-simoes.pt/page/ls-net>. Accessed: 2016-03-07.
- [2] Aplicação wtransnet cargo. <http://www.wtransnet.com/pt/app-bolsa-cargas/>. Accessed: 2016-03-07.
- [3] Body-parser source module. <https://github.com/expressjs/body-parser>. Accessed: 2016-06-15.
- [4] Express. fast, unopinionated, minimalist web framework for node.js. <https://expressjs.com/>. Accessed: 2016-06-15.
- [5] File system module. <https://nodejs.org/api/fs.html>. Accessed: 2016-06-15.
- [6] Jsonwebtoken source module. <https://github.com/auth0/node-jsonwebtoken>. Accessed: 2016-06-15.
- [7] Morgan source module. <https://github.com/expressjs/morgan>. Accessed: 2016-06-15.
- [8] Multiparty source module. <https://github.com/expressjs/node-multiparty>. Accessed: 2016-06-15.
- [9] Noticia tvi24, transportes de mercadorias. <http://www.tvi24.iol.pt/economia/transportes/desde-2012-fecharam-3-000-empresas-de-transporte-de-mercadorias>. Accessed: 2016-03-07.
- [10] Pg source module. <https://github.com/brianc/node-postgres>. Accessed: 2016-06-15.
- [11] Android.com. Meet android studio. <https://developer.android.com/studio/intro/index.html>. Accessed: 2016-06-15.

- [12] Nick Sears Andy Rubin, Rich Miner and Chris White. Ther history of android. <http://visual.ly/history-android>, 2003. Accessed: 2016-06-15.
- [13] Satya Komatineni DAve MacLean and Grant Allen. *Pro Android 5*. Apress, 2015.
- [14] Developer.Android.com. Activities. <https://developer.android.com/guide/components/activities.html>. Accessed: 2016-06-15.
- [15] Developer.Android.com. AsyncTask. <https://developer.android.com/reference/android/os/AsyncTask.html>. Accessed: 2016-06-15.
- [16] Developer.Android.com. Intents and intent filters. <https://developer.android.com/guide/components/intents-filters.html>. Accessed: 2016-06-15.
- [17] Multicargo Transportes e Logistica. *Impacto da Contratação de uma empresa de Transportese Logística*. 2015.
- [18] Economias.pt. Importações e exportações em portugal. <https://www.economias.pt/importacoes-e-exportacoes-em-portugal/>. Accessed: 2016-03-07.
- [19] Eurotransporte. Luís simões lança aplicação móvel lsnet. <http://eurotransporte.pt/noticia/1677/luis-simoes-lanca-aplicacao-movel-lsnet/>, 2015. Accessed: 2016-03-07.
- [20] Donn Felker and Joshua Dobbs. *Android Application Development for Dummies*. Wiley Publishing, Inc., 2011.
- [21] Grazielle Goulart. Tipos de transportes e cargas. <http://transportesdecarg.blogspot.pt/2012/06/tipos-de-transportes-x-cargas.html>. Accessed: 2016-03-07.
- [22] The PostgreSQL Global Development Group. About postgresql. <https://www.postgresql.org/about/>. Accessed: 2016-06-15.
- [23] Dickson S. Guedes. O que o postgresql tem a nos oferecer? <http://guedesoft.net/blog/2009/01/21/o-que-o-postgresql-tem-a-nos-oferecer/>. Accessed: 2016-06-15.

- [24] Jakob Jenkov. N tier architecture. <http://tutorials.jenkov.com/software-architecture/n-tier-architecture.html>, 2014. Accessed: 2016-06-15.
- [25] JetBrains.com. Webstorm features. <https://www.jetbrains.com/webstorm/features/>. Accessed: 2016-06-15.
- [26] Lorena Rodrigues dos Santos Pereira e Stephanie Cristhyne Araujo da Silva Ludmyla Barbosa. Transportes de carga. <http://meios-de-transporte.info/transporte-de-carga.html>. Accessed: 2016-03-07.
- [27] Tony Marston. What is the 3-tier architecture? <http://www.tonymarston.net/php-mysql/3-tier-architecture.html>, 2012. Accessed: 2016-06-15.
- [28] Netmarketshare. Mobile/tablet operating system market. Accessed: 2016-06-15.
- [29] Nodejs.org. Node.js foundation survey results. <https://nodejs.org/en/foundation/case-studies/>. Accessed: 2016-06-15.
- [30] Lauren Orsini. What you need to know about node.js. <http://readwrite.com/2013/11/07/what-you-need-to-know-about-nodejs/>. Accessed: 2016-06-15.
- [31] Pgadmin.org. pgadmin introduction. <https://www.pgadmin.org/>. Accessed: 2016-06-15.
- [32] Mr. Nico Reed. How do i create a http server? <https://docs.nodejitsu.com/articles/HTTP/servers/how-to-create-a-HTTP-server/>. Accessed: 2016-06-15.
- [33] Adriano Gomes Helder Marques Mark Macedo Ricardo Teixeira Vitor Silva. Transportes. <http://pt.slideshare.net/francisogam/vantagens-e-desvantagens-do-transporte-maritimo>. Accessed: 2016-03-07.
- [34] Luis Simões. Comunicado de imprensa. <http://www.luis-simoes.pt/uploads/20150907%20PR%20App%20LSNet%20.pdf>. Accessed: 2016-03-07.
- [35] Luis Simões. Lsnet, google play store. <https://play.google.com/store/apps/details?id=com.innovagency.luissimoes>. Accessed: 2016-03-07.
- [36] Stefan Tilkov. Uma rápida introdução ao rest. <https://www.infoq.com/br/articles/rest-introduction>, 2008. Accessed: 2016-06-15.

- [37] JORNAL DOS TRANSPORTES. Luís simões apresenta aplicação móvel lsnet. <http://jornaldotransportes.pt/luis-simoes-apresenta-lsnet/#prettyPhoto>, 2015. Accessed: 2016-03-07.
- [38] JORNAL DOS TRANSPORTES. Wtransnet lança aplicação móvel de pesquisa e gestão de cargas. <http://jornaldotransportes.pt/wtransnet-lanca-aplicacao-movel-de-pesquisa-e-gestao-de-cargas/>, 2015. Accessed: 2016-03-07.
- [39] Tutorialspoint. Android - architecture. http://www.tutorialspoint.com/android/android_architecture.htm. Accessed: 2016-06-15.
- [40] Deyve Vieira. Postgresql introduction and concepts. <http://www.tuxlinux.com.br/e-book-gratuito-postgresql-introduction-and-concepts/>. Accessed: 2016-06-15.
- [41] Techlab Wtransnet. Wtransnet cargo, google play store. https://play.google.com/store/apps/details?id=com.wtransnet.app&hl=pt_PT. Accessed: 2016-03-07.